MRPC: Maximizing Network Lifetime for Reliable Routing in Wireless Environments

Archan Misra
IBM T.J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532, USA
Email: archan@us.ibm.com

Suman Banerjee
Department of Computer Science
University of Maryland
College Park, MD 20742, USA
Email: suman@cs.umd.edu

***Abstract*— We propose MRPC, a new power-aware routing algorithm for energy-efficient routing that increases the operational lifetime of multi-hop wireless networks. In contrast to conventional power-aware algorithms, MRPC identifies the capacity of a node not just by its residual battery energy, but also by the expected energy spent in reliably forwarding a packet over a specific link. Such a formulation better captures scenarios where link transmission costs also depend on physical distances between nodes and the link error rates. Using a max-min formulation, MRPC selects the path that has the largest packet capacity at the 'critical' node (the one with the smallest residual packet transmission capacity). We also present CMRPC, a conditional variant of MRPC that switches from minimum energy routing to MRPC only when the packet forwarding capacity of nodes falls below a threshold. Simulation-based studies have been used to quantify the performance gains of our algorithms.**

I. Introduction

Energy efficient routing algorithms are very important in wireless multi-hop networks, where communication costs (transmission power) are usually more expensive than computing costs, and where the constituent nodes have batteries with limited energy. Several energy-aware routing protocols (e.g., [1], [2]) define the link cost as a function of the power required to transmit a packet across the link, and then employ minimum cost routing algorithms to determine the "minimum total transmission energy" route from a source to the destination.

In many wireless ad-hoc scenarios, however, the metric of actual interest is not the transmission energy of individual packets, but the *total operational lifetime of the network.* To avoid the extinction of nodes due to exhaustion of their battery power, *power-aware* routing algorithms try to ensure an equitable distribution of the transmission costs among the constituent nodes. It is easy to see that the two routing objectives can be mutually contradictory. For example, if several minimum energy routes have a common host, the battery power of that host will be exhausted quickly. From a conceptual standpoint, power-aware routing algorithms (e.g., [3], [4]) attempt to distribute the transmission load over the nodes in a more egalitarian fashion, even if such distribution drives up the overall energy expenditure. Such algorithms do not, however, consider the possibility that different links can have different transmission costs. However, in our prior work [6], we have shown that, for reliable communication, link characteristics (e.g., the packet error rate of the link) can significantly affect the energy requirements for packet transmission over that link.

The main contribution of this paper is in showing how power-aware routing protocols must not only be based on node specific parameters (e.g. residual battery energy of the node), but must also consider the link specific parameters (e.g. channel characteristics of the link) as well, to increase the operational lifetime of the network. We present a new power-based route selection algorithm called the Maximum Residual Packet Capacity (MRPC). This algorithm can be easily integrated into a variety of ad-hoc routing protocols. MRPC is conceptually similar to the MMBCR algorithm [4] in that, at any point in time, it tries to select the route that maximizes the residual capacity currently available at the most critical node (the one with the least residual capacity). MRPC accommodates scenarios where the nodes can adjust their transmission power dynamically (based on the distance between the nodes), and also incorporates the effect of link layer error rates and consequent packet re-transmissions.

In MRPC, the cost of choosing a particular link at any instant is defined as the *idealized maximum number of packets* (or bits to be more general) that can be transmitted by the transmitting node over the specific link, assuming the complete absence of any other cross traffic at that node. We use simulation studies to show how MRPC leads to superior performance (longer network lifetimes or larger number of successfully transmitted packets) than alternative suggested algorithms, due to its accommodation of variability in the transmission energy and the packet error probabilities in the link cost.

Since minimum-energy routes are more energy efficient, a conditional variant of the MMBCR algorithm was also proposed in [4]. In this scheme (called Conditional MMBCR, or CMMBCR) minimum energy routes were chosen till the residual battery power of constituent nodes of the routes fell below a specified threshold level. Once this threshold level is crossed, routes are chosen using the MMBCR algorithm, which equitably distributes the battery consumption among the different nodes thus protecting against the early exhaustion of a few nodes. We also present the conditional analogue of MRPC, the Conditional MRPC (CMRPC) algorithm and then evaluate its performance, vis-a-vis the CMMBCR algorithm. CMRPC performs minimum energy routing (using the link cost formulation for reliable transmissions as presented in [6]) as long as the remaining battery power at the constituent nodes lie above a specified threshold. Beyond this point, CMRPC switches to the MRPC-based max-min path selection algorithm.

II. Prior and Related Work

Most energy-aware routing protocols are based on the observation that the signal attenuation (and hence, the required transmission power level at the sender) is $\propto d^K$ $(K \geq 2)$, where d is the transmission distance. Accordingly, an energy-aware routing algorithm would select a route comprising multiple short-distance hops over another one with a smaller hop count but larger hop distances. PAMAS [1] was developed as a minimum energy routing protocol, where the transmission power ($\propto d^K$) between two nodes was used to represent the cost associated with the corresponding link. Using Dijkstra's shortest path algorithm to compute the minimum cost path, then yields the minimum total power route. Bias towards smaller hops typically led to the selection of paths with a very large hop count. In [2], authors suggested the inclusion of the energy expended in receiving packets in the packet forwarding cost. A modified form of the Bellman-Ford algorithm was then used to derive paths, which were, consequently, shorter than those computed by [1]. The PARO routing protocol [5] has recently been proposed for such network situations where the transmission energy is variable; the protocol essentially allows an intermediate node to insert itself in the routing path if it detects potential savings in the transmission energy.

All these minimum total energy protocols, however, ignore the costs of potential re-transmissions across an error-prone link. In [6], we showed how such a formulation can lead to sub-optimal path choices, since it ignores this potential need for multiple re-transmissions to achieve reliable packet delivery in the presence of link errors. Wireless links, typically perform link-layer re-transmissions, and therefore choosing a path with a very large number of short hops can be counter-productive. We make the following observation:

> *As the number of hops is increased, the resultant increase in the total number of re-transmissions, needed to ensure reliable packet delivery over the large number of hops, can negate the reduction achieved using short-range hops.*

Some prior work has also addressed the issue of increasing the operational lifetime of a multi-hop wireless network. The use of the remaining battery power of an node as a metric for energy-efficient routing was reported in [3]. It suggested using a battery cost function $f(B_i) = \frac{1}{B_i}$ where B_i is the residual battery capacity of node i. By using $f(.)$, the energy-aware path selection was formulated as a minimum cost path calculation, where the cost of a specific path was the sum of the individual battery cost functions of the constituent nodes; the cost of a path P was given by:

$$C_P = \sum_{node\ i\ \epsilon\ P} f(B_i)$$

The MMBCR algorithm presented in [4] is most closely related with the MRPC formulation. MMBCR, like MRPC, uses a min-max route selection technique, with the algorithm choosing that route that has the largest value for its most critical ("bottleneck") node–i.e., the node with the least residual battery capacity. Mathematically speaking, MMBCR associates with a specific path P a cost metric C_P given by

$$C_P = \min_{\text{node } i \text{ lies on route P}} \{B_i\}$$

where B_i is the residual battery power level on node i. The path then selected P_{MMBCR} is given by:

$$P_{MMBCR} = \arg \max_{P} \{C_P\}$$

The Conditional MMBCR algorithm (CMMBCR) was also presented in [4]. Since the MMBCR algorithm never tries to minimize the total transmission energy along a path (it is *always* concerned with spreading the transmission cost evenly among available nodes), it can lead to overall higher energy consumption and consequently, a reduction of the average node lifetime. To counteract this, the CMMBCR algorithm uses the minimum energy path initially, as long as the battery power level on all the nodes in the selected path lie above a certain threshold γ. Once one or more of nodes on all possible paths falls below this battery protection threshold, the algorithm switches to the MMBCR mode. However, unlike MRPC (and CMRPC), MMBCR (and CMMBCR) does not take into account the possibility that links may have widely varying transmission energy costs and link error probabilities.

III. The MRPC Algorithm

Our previous discussion shows that selecting the path with the least transmission energy for reliable communication may not always maximize the lifetime of the ad-hoc network. Moreover, since the actual drain on a node's battery power will depend on the number of packets forwarded by that node, it is difficult to predict the optimal routing path unless the total size of the packet stream is known during path-setup. Accordingly, MRPC works on selecting a path, given the current battery power levels at the constituent nodes, that maximizes *the total number of packets that may be ideally transmitted over that path*, assuming that all other flows sharing that path do not transmit any further traffic.

To formalize this concept, assume that the residual battery power at a certain instance of time at node i is B_i. Also, let us assume that the transmission energy required by node i to transmit a packet over link (i, j) to node j is $E_{i,j}$. (We shall later discuss the various formulations for $E_{i,j}$). Let the source and destination nodes for a specific session (route) be S and D respectively.

If the route-selection algorithm then selects a path P from S to D that includes the link (i,j), then the maximum number of packets that node i can forward over this link is clearly $\frac{B_i}{E_{i,j}}$. Accordingly, we can define a **node-link metric**, $C_{i,j}$ for the link (i,j) as :

$$C_{i,j} = \frac{B_i}{E_{i,j}} \tag{1}$$

The key point in this formulation is that the cost metric includes both a node-specific parameter (the battery power) and a link-specific parameter (the packet transmission energy for reliable communication across the link).

Clearly, the maximum "lifetime" of the chosen path P, defined by the maximum number of packets that may be potentially forwarded between S and D using path P, is determined by the *weakest intermediate node*– one with the smallest value of $C_{i,j}$. Accordingly, the "maximal lifetime" associated with route P is seen to be:

$$Life_P = \min_{(i,j)\epsilon P} \{C_{i,j}\} \tag{2}$$

The MRPC algorithm then selects the route $P_{candidate}$ that maximizes the "maximal lifetime" of communication between S and D. Formally, the chosen route is such that:

$$P_{candidate} = \arg\ \max\{Life_P | P\epsilon\ all\ possible\ routes\} \tag{3}$$

While the computed route may be optimal at the time of computation, the random traffic patterns will potentially make the currently selected paths sub-optimal at some point in the future. Thus, MRPC is really a route selection algorithm; a routing protocol that uses MRPC for multi-hop wireless networks will include mechanisms for periodic and distributed route computation.

A. Potential De-Centralized Implementation

Given the cost and lifetime formulations for MRPC (Equations (1) and (2)), it is then easy to use a modified version of Dijkstra's minimum cost algorithm for decentralized route computation. While $Life_P$ is not an additive function of the individual node-link costs, it can be computed over a path by applying the MIN operator in an iterative fashion.

To apply Dijkstra's algorithm for determining the minimum-cost path, we define the distance metric from any node to the given destination as the value of $Life_P$ over the optimal path from that node to D. Now consider a node A that sees advertisements from three "neighbors", X,Y and Z, with corresponding distance metrics $Life_X$, $Life_Y$ and $Life_Z$ for a given destination D. Node A can then compute the best path to D (using its optimal neighbor) by using the following simple algorithm:

1) For each of the neighboring nodes ($j\ \epsilon\ \{$X,Y,Z$\}$), compute the link cost $C_{A,j}$ using Equation (1).
2) For each of the neighboring nodes ($j\epsilon\{$X,Y,Z$\}$), compute the potential new value of $Life_{pot}$ using

$$Life_{pot}(A,j) = \min\ \{C_{A,j}, Life_j\}$$

3) Select as the next-hop neighbor towards D that node which results in the maximum value of $Life_{pot}$, i.e., chose node k such that

$$k = \arg \max_{j\epsilon\{X,Y,Z\}} \{Life_{pot}(A,j)\}$$

It is easy to see that using this recursive formulation allows all nodes in the ad-hoc network to iteratively build their optimal route towards a specific destination D. The distance-vector formulation presented here can easily be incorporated in protocols, such as TORA [7], AODV [8], that are specifically designed for ad-hoc mobile environments. Indeed, the intent of this paper is not to indicate the choice of a specific routing protocol, but to define a set of power-aware metrics for use by a protocol during route selection.

B. Applying MRPC to Energy-Aware Cost Formulations

The basic MRPC formulation for power-aware routing does not specify the value of the transmission energy cost associated with a specific link– Equation (1) is expressed in terms of an abstract value $E_{i,j}$. Accordingly, by specifying different forms of $E_{i,j}$, it is possible to tailor the MRPC mechanism for specific technologies and/or scenarios.

For example, for radio technologies where the transmission power is a constant, the energy involved in a single packet transmission attempt, $T_{i,j}$, is a constant for all (i,j) and is independent of the distance between neighboring nodes i and j. If the transmitter is, however, capable of dynamically adjusting its power based on the link distance, $T_{i,j}$ will typically be $\propto D_{i,j}^K$, where $D_{i,j}$ is the distance between nodes i and j.

In [6], we showed why a routing algorithm for reliable packet transfer should include the link's packet error probability in formulating the transmission energy cost. By ignoring the packet error probability, the link cost concentrates (wrongly) only on the energy spent in transmitting a single packet; the correct metric is the *effective* packet transmission energy for reliable transmission, which includes the energy spent in one or more re-transmissions that might be necessary in the face of link errors. [6] suggested a transmission energy metric of the form

$$E_{i,j} = \frac{T_{i,j}}{(1-p_{i,j})^L}$$

where $p_{i,j}$ is the link's packet error probability, $L\ \epsilon\ \{1, 2, \ldots\}$. In fact, in [6] we showed that the presence of hop-by-hop re-transmissions (a reliable link layer) implied that $L = 1$; in the absence of hop-by-hop re-transmissions (i.e. only re-transmissions are performed end-to-end), the transmission cost is well approximated by $L\ \epsilon\ \{3, \ldots, 5\}$.

It should thus be clear that MRPC degenerates to MMBCR *only if* all nodes are incapable of dynamically adapting their power based on the transmission range, and *only if* all links have the same intrinsic error rates. In all other cases, MRPC makes a more intelligent choice, since it takes into consider the potential variability in the energy needed for reliable packet transfer.

For the simulation results reported later, we concentrate on the typical scenario, where wireless links are capable of link-level re-transmissions of corrupted frames, and with nodes capable of dynamically adjusting their power. Accordingly, we have

$$E_{i,j} = \frac{D^K}{1 - p_{i,j}} \tag{4}$$

C. CMRPC

The CMRPC algorithm is the MRPC equivalent of the CMMBCR algorithm presented in [4]. The CMMBCR algorithm is based on the observation that using residual battery energy as the sole metric throughout the lifetime of the ad-hoc network can actually lower the overall lifetime, since it never attempts to minimize the total energy consumption. Accordingly, the CMMBCR algorithm uses regular minimum-energy routing as long as there is even one candidate path, where the remaining battery power level in all the constituent nodes lies above a specified threshold γ. When no such path exists, CMMBCR switches to MMBCR, i.e., it picks the path with the maximum residual capacity on the "critical node".

Our CMRPC algorithm differs from CMMBCR in that the cost-functions at all times include the link-specific parameters (e.g. error rates) as defined in [6]. The algorithm can thus be specified as follows. Let Ψ be the set of all possible paths between the source S and destination D. At any point of time, let Ω represent the set of paths such that:

$$Life_P \geq \gamma \;\; \text{for any route } P \;\epsilon\; \Omega$$

i.e, Ω represents the set of paths whose most critical nodes have a lifetime greater than a specified threshold. The routing scheme thus consists of the following actions:

a) If $\Omega \cap \Psi \neq \Phi$ (there are one or more paths with $Life$ greater than the threshold, the algorithm selects a path $\bar{P} \;\epsilon\Omega$ that minimizes the total transmission energy for reliable transfer, i.e.,

$$\bar{P} = \underset{P\epsilon\Omega}{\arg\min} \{ \sum_{(i,j)\;\epsilon\; P} E_{i,j} \} \tag{5}$$

b) Otherwise, switch to the MRPC algorithm, i.e., select $\bar{P}$ such that

$$\bar{P} = \arg\max\{Life_P | P\epsilon\Psi\}$$

The threshold γ is a parameter of the CMRPC algorithm– a lower value of γ implies a smaller protection margin for nodes nearing battery power exhaustion. Accordingly, the performance of the CMRPC algorithm will be a function of γ.

IV. Performance Studies

In this section, we report on extensive simulation studies to understand the performance benefits and tradeoffs associated with the MRPC and the CMRPC routing algorithms. We compare the performance of 6 different routing schemes:

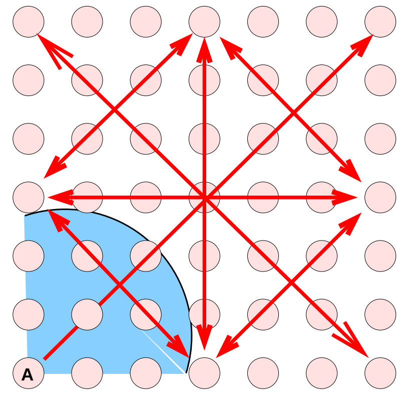

Fig. 1. The 49-node topology. The shaded region marks the maximum transmission range for the corner node, A when R=2.9.

a) *Min-Hop Routing*: This is the conventional "energy-unaware" Internet routing algorithm, where each link is assigned an identical cost.

b) *Min-Energy Routing*: This algorithm, based on the formulation in [6], simply selects the path corresponding to the minimum packet transmission energy for reliable communication, without considering the battery power of individual nodes.

c) *MMBCR*: This power-aware routing algorithm, described earlier, selects the path whose critical node has the highest residual battery energy.

d) *CMMBCR*: This algorithm, also described earlier, switches from minimum-energy paths to MMBCR when the residual battery energy falls below a specified threshold.

e) *MRPC*: Our algorithm always uses the path with the highest value of $Life_P$.

f) *CMRPC:* This conditional version of MRPC uses the min-energy algorithm as long as $Life_P$ associated with the chosen route P lies above a specified threshold; once the critical link-node cost function falls below this value, the algorithm switches to MRPC-based routes.

A. Simulation Model

For our experiments, we used different topologies having up to 100 nodes randomly distributed over on a square region, to study the effects of various schemes on energy requirements and throughputs achieved. In this section, we discuss in detail results from one representative topology, where 49 (static) nodes were distributed over a 7X7 unit grid, equi-spaced 1 unit apart (Figure 1). The corner nodes and the mid-points of each side of the rectangular grid were chosen as traffic sources and destinations–the bold lines in Figure 1 show the session end-points. Each (source, destination) pair had 2 simultaneous sessions activated in the opposite direction, giving rise to a total of 16 different sessions. For the results reported here, each session consisted of a UDP traffic generated by a CBR source whose inter-packet gap was distributed uniformly between $0.1 - 0.2$ secs. The error rate on each link was independently distributed

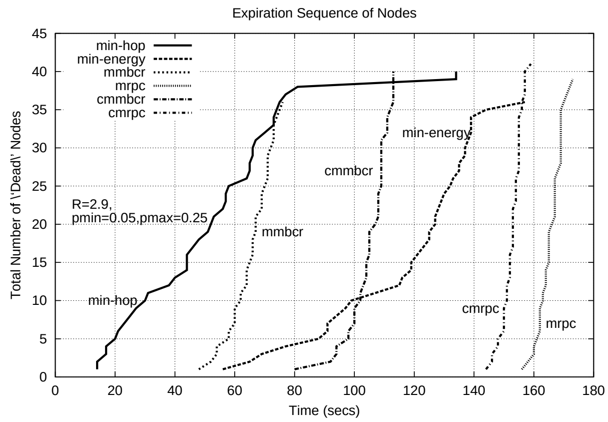

Fig. 2. Expiration Sequence for Different Algorithms, R=2.9

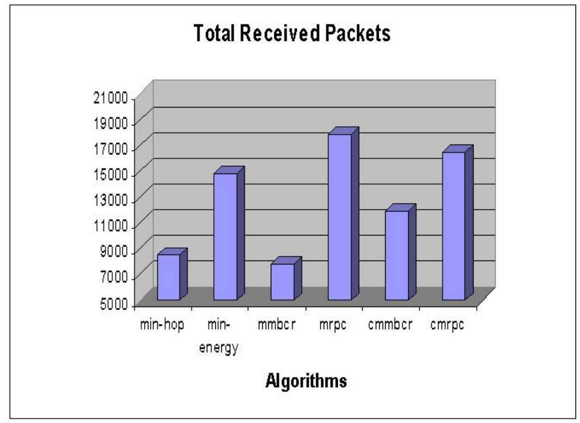

Fig. 3. Total Packet Throughput (UDP Sources), R=2.9

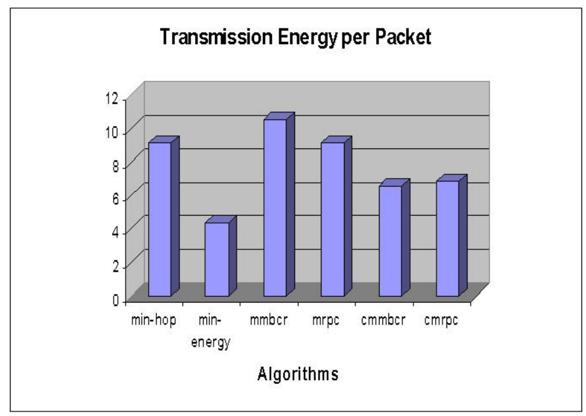

Fig. 4. Avg. Transmission Energy per Received Packet (UDP Sources), R=2.9

uniformly between $(0.05, p_{max})$; we experimented with varying values of p_{max}. Routes were recomputed at 2 second intervals.

Whenever nodes died (when its battery power gets completely drained) during the course of a simulation, our simulation code would check whether the graph became partitioned. The simulations were run until each of the 16 sessions failed to find any route from their source to the corresponding destination. To avoid the termination of a simulation due to battery power exhaustion at source or destination nodes, all source and sink nodes were configured to have 'infinite' power resources. All the other 'intermediate' nodes were configured with identical initial battery power levels.

To study the performance of the various algorithms, we performed experiments where the maximum transmission radius, R, of each node was varied. Figure 1 shows the set of neighboring nodes for a corner node when the transmission radius is set to 2.9. We note the expiration sequence, as well as the node expiry times, for each simulation. The *expiration sequence* (sorted in ascending order of the expiration times) provides a useful indicator of how each algorithm affects the lifetime of the individual nodes, and the entire network. In addition to the expiration sequence, we also calculate the *total packet throughput* by counting the total number of packets successfully received at the destination nodes, and the *energy costs per packet* by dividing the total energy expenditure by the total packet throughput. Except for the expiration sequences, all other metrics were obtained by averaging over multiple runs.

B. Results for R= 2.9

Figures 2, 3 and 4 plot the node expiration sequence, the total packet throughput and the effective energy per packet respectively for all the routing algorithms when the transmission radius, R, was set equal to 2.9 units and $p_{max} = 0.25$. In this case, the corner node has a choice between 8 one-hop neighbors. The protection threshold was set to 50% of the total battery capacity for CMMBCR and 50% of the average initial packet capacity for CMRPC. We can see that, as expected, the min-hop algorithm performs the worst, since it not only fails to balance the workload among the intermediate nodes, but also uses large-distance hops and consequently larger transmission energy. In contrast, while the min-energy algorithm does use smaller individual hops, it is also susceptible to high variability in the expiration sequence. The plot effectively demonstrates the superior performance of MRPC over the MMBCR algorithm, which does not consider the fact that different links have dramatically different transmission energy per packet. While the use of MMBCR leads to the expiration network by ~ 80 secs, the MRPC algorithm is able to ensure packet transfer (by at least one session) till $\sim\ 170$ secs. The figure also demonstrates the relative performance benefits of the conditional variants of MMBCR and MRPC. CMMBCR performs much better than MMBCR, since during the initial "minimum-energy routing" phase, it considers the differential transmission energy consumed by

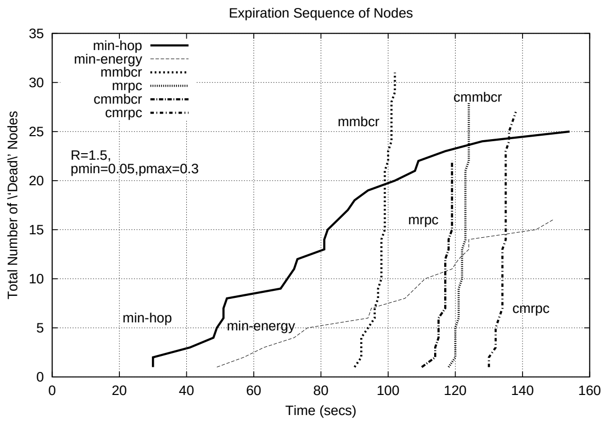

Fig. 5. Expiration Sequence for Different Algorithms, R=1.5

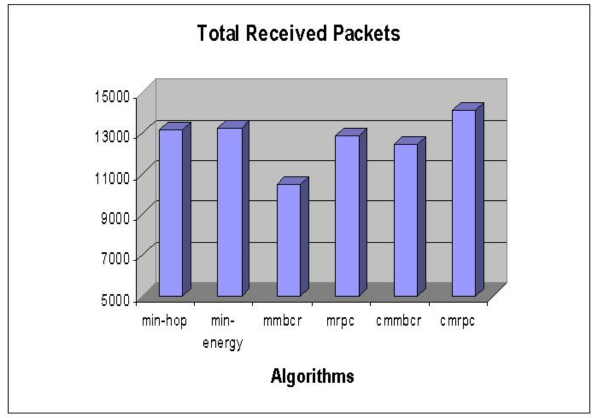

Fig. 6. Total Packet Throughput (UDP Sources), R=1.5

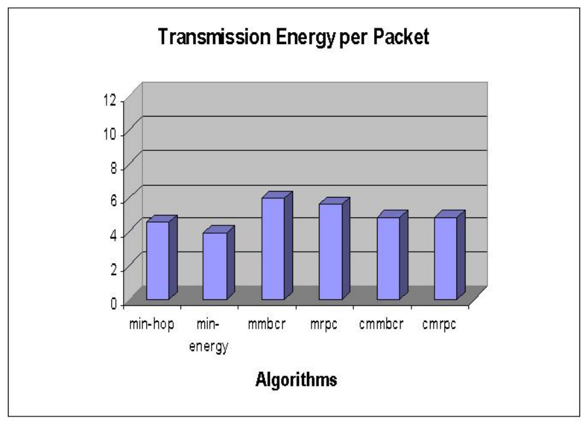

Fig. 7. Avg. Transmission Energy per Received Packet (UDP Sources), R=1.5

different links. In contrast, CMRPC (for this choice of γ) does not outperform MRPC; in fact, CMRPC and MRPC do not exhibit significant differences in their expiration sequence. This occurs because, unlike MMBCR, the reliable packet transmission cost based formulation [6] of the MRPC cost implicitly results in the selection of links that spent less transmission energy per packet.

The performance variation among the algorithms can be observed more clearly in Figures 3 and 4. While the min-energy algorithm obviously results in the lowest effective energy per packet, it also results in a much smaller total packet throughput. In contrast to MMBCR, MRPC is not only able to transmit a much larger number of packets but also at a lower per-packet energy consumption. Similarly, CMRPC outperforms CMMBCR in both the total packet throughput as well as the energy efficiency, although the differences are less dramatic due to the presence of a common 'minimum total energy' phase in both the algorithms.

C. Results for R=1.5

The dramatic improvement offered by MRPC and its conditional variant in the previous sub-section is partially explained by observing that a larger value of R implicitly results in a larger variation of the packet transmission costs for reliable communication among the various available links at any node. MRPC is, however, better than MMBCR even in situations where the distance variation among the candidate links is not very dramatic; even if the transmission energy for a single packet, $T_{i,j}$, was identical on all links, MRPC should perform better by selecting links with lower error rates and consequently, smaller energy expenditure on packet re-transmissions. To investigate this behavior, we performed simulations where the maximum transmission radius, R, equals 1.5; in this case, the corner nodes have only 3 candidate neighbors. Figures 5,6 and 7 plot the various metrics of interest for the candidate algorithms in this scenario, with $p_{max} = 0.3$ and γ for CMRPC set to 75% of the initial packet capacity. Once again, it can be seen that MRPC/CMRPC outperform the equivalent version of MMBCR. By selecting paths with smaller error rates, MRPC and CMRPC are able to not only able to significantly delay the onset of node expiration significantly, but can also achieve significantly larger packet throughput in a more energy-efficient manner. In this case, however, CMRPC performs better than MRPC in that it leads to longer network lifetime and a greater value of the total packet throughput while achieving a lower energy expenditure per packet. While the min-hop and minimum-energy algorithms outperform the power-aware algorithms in the energy and throughput metrics, they clearly result in a much higher variability in the node lifetimes. In networks where sessions are dynamically generated between a random pair of nodes, such early expiration of battery power is clearly undesirable. In fact, the goal of power-aware algorithms is precisely to trade off the energy efficiency along a set of specific paths for greater longevity of all the network nodes.

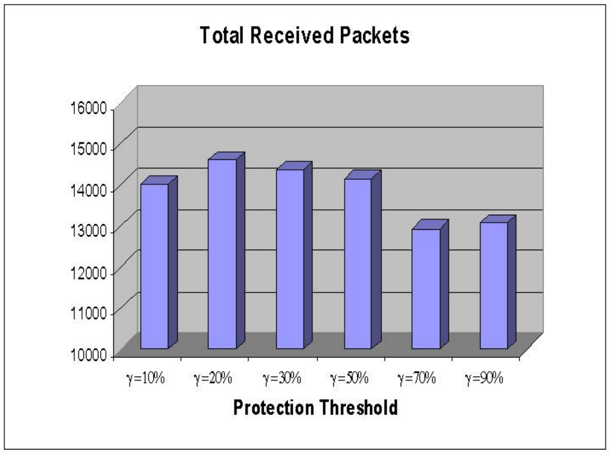

Fig. 8. CMRPC: Total Packet Throughput vs. γ

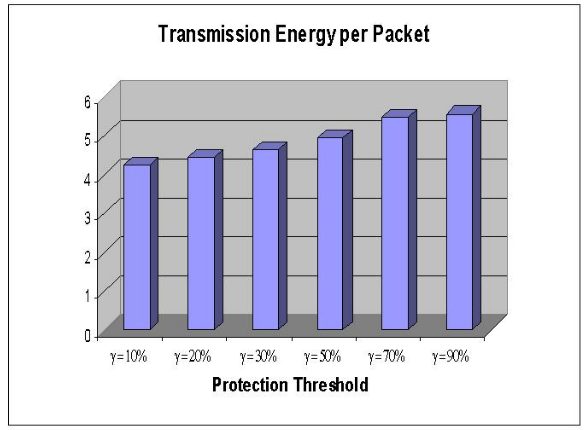

Fig. 9. CMRPC: Avg. Transmission Energy per Received Packet vs. γ

The above figures also show that the relative performance of MRPC and CMRPC depend on the choice of the threshold γ. Unlike the MMBCR case, CMRPC does not always outperform MRPC. Indeed, if γ is close to 100%, CMRPC degenerates to MRPC; on the other hand, if γ is close to 0%, CMRPC degenerates to min-energy routing. Figures 8 and 9 show the total network throughput and energy per transferred packet as the CMRPC protection threshold γ is varied. Clearly, the average energy per packet increases with increasing γ, as CMRPC performs minimum-energy routing for a smaller duration. On the other hand, the total network throughput is maximized at an intermediate value for γ (around 20% in Figure 8)–while smaller values (longer min-energy routing) lead to higher variability in the expiration times, larger values fail to exploit minimum-energy paths even if the residual battery capacities are sufficiently large.

V. Conclusion

In this paper, we have presented two power-aware algorithms for energy-efficient routing in ad-hoc wireless networks. In contrast to previous power-aware algorithms, our proposed approaches do not base their routing decisions (and link costs) on a function of the battery power alone. Rather, they also consider the fact that different links require different transmission powers, and also have different impacts on reliable packet transfers due to differences in their packet error rates. While the MRPC algorithm employs a min-max formulation at all times, the CMRPC variant switches from minimum-energy routing to MRPC only when the forwarding capacity of intermediate nodes drops below a threshold value.

Our simulation experiments confirm that MRPC and CMRPC outperform their MMBCR counterparts, primarily by exploiting the knowledge that different links can impose significantly different transmission costs during reliable packet forwarding. In comparison to MMBCR, MRPC is able to not only significantly extend the lifetime of the network, but is also able to transmit a significantly greater number of packets at a higher energy efficiency (smaller reliable transmission cost per packet). We are currently working on incorporating the MRPC metrics in an ad-hoc routing protocol and then performing energy-based studies in mobile, ad-hoc scenarios. Unlike our current idealized study, this future work will also include the energy overheads associated with signaling and mobility management in such ad-hoc environments.

References

[1] S. Singh and C.S. Raghavendra, “PAMAS-Power Aware Multi-Access Protocol with Signaling for Ad Hoc Networks”, *ACM Communication Review*, July 1998.

[2] K. Scott and N. Bambos, “Routing and Channel Assignment for Low Power Transmission in PCS”, *Proceedings of ICUPC’96*, October 1996.

[3] S. Singh, M. Woo and C.S. Raghavendra, “Power-Aware Routing in Mobile Ad Hoc Networks”, *Proceedings of Mobicom’98*, October 1998.

[4] C.K. Toh, H. Cobb and D. Scott, “Performance Evaluation of Battery-Life-Aware Routing Schemes for Wireless Ad Hoc Networks”, *Proceedings of IEEE ICC’01*, June 2001.

[5] J. Gomez, A. Campbell, M. Naghshineh and C. Bisdikian, “PARO: A Power-Aware Routing Optimization Scheme for Mobile Ad hoc Networks”, *draft-gomez-paro-manet-00.txt*, IETF, Work in Progress, February 2001.

[6] S. Banerjee and A. Misra, “Minimum Energy Paths for Reliable Communication in Multi-hop Wireless Networks”, *CS-TR 4315, Technical Report, Department of Computer Science, University of Maryland, College Park*, December 2001.

[7] V. Park and S. Corson, “Temporally-Ordered Routing Algorithm (TORA) Version 1: Functional Specification”, draft-ietf-manet-tora-spec-04.txt, *IETF*, Work in Progress, July 2001.

[8] C. Perkins, E. Royer and S. Das, “Ad Hoc On Demand Distance Vector (AODV) Routing”, draft-manet-ietf-aodv-8.txt, *IETF*, Work in Progress, March 2001.